UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE 2Bh

Thursday 30 May 2002

14:00 to 16:30

Examiners

D.K. Arvind (Chair)

R. Dyckhoff (External)

INSTRUCTIONS TO CANDIDATES

Section A questions are worth 10 marks each.

Section B questions are worth 30 marks each.

Answer all FOUR questions in Section A.

Answer any TWO of the three questions in Section B.

If you are in the third or later year of your degree course, please tick the box on the front cover of the script book.

Section A

1. (a) Explain briefly the difference between a *public* and a *private* method in a Java class.

[2 marks]

(b) Write complete Java code for a method `maxpos` that takes an array of integers and returns the largest positive value in the array. If the array is null, or empty, or contains only negative values, then the method should return zero. It must never throw an exception.

[8 marks]

2. (a) Recall that a *hash function* for storing keys in a hash table of size N assigns to every key an integer in the range $0, \ldots, N-1$ called the *hash value* of the key.

Briefly explain what a *hash code* and a *compression map* are and how they are composed to form a hash function.

[3 marks]

Suppose now that we want to store items whose keys are words over the alphabet $\{a, b, c\}$ in a hash table of size $N = 20$.

Let f be the mapping from a, b, c to the integers defined by

$$f(a) = 1, \quad f(b) = 2, \quad f(c) = 3.$$

As our hash code, we use the mapping F from $\{a, b, c\}^*$ to the integers defined by

$$F(x_1 \ldots x_n) = f(x_1) + \cdots + f(x_n).$$

For example, $F(abc) = 1 + 2 + 3 = 6$.

As our compression map, we use the mapping G from the integers to the integers defined by

$$G(k) = (7 \cdot k + 3) \bmod 20.$$

For example, $G(6) = 5$.

(b) Compute the hash values of the keys a, bac, $ccccccc$, $abcccccc$.

[4 marks]

(c) Describe an infinite sequence of keys that all get the same hash value.

[3 marks]

3. The following SQL commands were used to create the tables `EMP`, and `DEPT` for the example database that you used in the course:

```
CREATE TABLE EMP
  (EMPNO INTEGER NOT NULL PRIMARY KEY,
   ENAME CHAR(10) NOT NULL,
   JOB CHAR(9),
   MGR INTEGER,
   HIREDATE DATE,
   SAL INTEGER,
   COMM INTEGER,
   DEPTNO INTEGER NOT NULL);

CREATE TABLE DEPT
  (DEPTNO INTEGER NOT NULL,
   DNAME CHAR(20),
   LOC CHAR(20));
```

(a) Consider the following SQL query:

```
SELECT DNAME
FROM   DEPT X
WHERE  NOT EXISTS (SELECT *
                   FROM EMP
                   WHERE EMP.DEPTNO = X.DEPTNO)
```

Give a description, in English, of what information this query requests. Note that this question requires you to describe the *information* requested, rather than the SQL constructs used in the query.

[2 marks]

(b) Construct an SQL query which lists the names of all departments that do not have any employees earning commission. You should assume a `NULL` value in the `COMM` column indicates an employee who does not earn commission.

[4 marks]

(c) Construct an SQL query which lists the names of all employees who earn more than their manager.

[4 marks]

4. (a) Explain briefly the meaning of *dynamic method invocation* in an object-oriented programming language like Java.

[2 marks]

(b) Here are two Java class declarations.

```
class Bird {
  void a() { System.out.println ("a in Bird"); }
  void b() { System.out.println ("b in Bird"); }
}

class Parrot extends Bird {
  void a() { System.out.println ("a in Parrot"); }
}
```

Based on these classes, write down the output of the following code fragment.

```
Bird   x = new Bird();
Parrot y = new Parrot();
Bird   z = new Parrot();

x.a(); x.b(); y.a(); y.b(); z.a(); z.b();
```

There should be exactly six lines of output. Show which ones are due to *overriding* and which ones to *inheritance*.

[8 marks]

Section B

5. Let A be an array storing n integers. An *inversion* of A is a pair (i, j) of indices such that $i < j$ and $A[i] > A[j]$. For example, the inversions of $\langle 17, 24, 11, 20\rangle$ are $(0, 2), (1, 2), (1, 3)$.

   Intuitively, the number of inversions of an array is a measure for the number of entries that are "out of order".

   (a) Find all inversions of the arrays $\langle 23, 19, 11, 28, 22\rangle$ and $\langle 2, 3, 4, 5, 6, 1\rangle$.

   [6 marks]

   (b) Which array with elements from the set $\{1, \ldots, n\}$ has the most inversions? How many does it have?

   [6 marks]

   (c) Design an algorithm that gets as input an integer array A and that computes the number of inversions of A. The asymptotic worst-case time complexity of your algorithm should be $\Theta(n^2)$.

   (i) Give a description of your algorithm in pseudo-code.

   (ii) Analyse the worst-case time complexity of your algorithm and justify that it is $\Theta(n^2)$.

   [8 marks]

   (d) Let A be an array storing n integers, and let $0 \leq m < n$. Let

   - k_1 be the number of inversions of the array $A[0 \ldots m]$,
   - k_2 be the number of inversions of $A[m + 1 \ldots n]$,
   - ℓ be the number of pairs (i, j) such that $0 \leq i \leq m$, $m + 1 \leq j \leq n$, and $A[i] > A[j]$.

   What is the number of inversions of A?

   [4 marks]

   (e) Briefly describe the basic idea for an improved algorithm for computing the number of inversions of an integer array. (Note that the phrase 'improved algorithm' means here an alogrithm with asymptotic worst-case time complexity of $\Theta(n \cdot \lg(n))$.)

   [6 marks]

   Hint: Use the divide-and-conquer principle. Part (d) of this problem should help.

6. XSLT and CSS provide alternative approaches to viewing XML documents via the rendering engines built in to web browsers.

   (a) Describe the basic formatting model used by ordinary HTML web browsers. Describe how each of the following HTML elements is rendered in terms of the model: `P`, `DIV`, `H1`, `H2`, `STRONG`, `SPAN`.

   [6 marks]

   (b) Describe the basic processes involved in viewing an XML document using CSS. Explain how CSS exploits the formatting model you described in your answer to (a) above. Be sure to cover the three basic sources of information about the rendering of elements when CSS is involved.

   [6 marks]

   (c) Describe the basic processes involved in viewing an XML document using XSLT. Explain how XSLT exploits the formatting model you described in your answer to (a) above. Be sure to include a description of how an XML document is traversed by the XSLT processor, and the difference between online and offline processing.

   [8 marks]

   (d) With reference to your answers to (b) and (c) above, compare CSS and XSLT as approaches to viewing XML documents. Identify contrasting strengths and weaknesses, and suggest requirements which might favour one approach or the other.

   [10 marks]

7. (a) All Java implementations provide support for multi-threaded programming. However, different virtual machines may use different methods to switch between threads. Explain the difference between *preemptive* and *non-preemptive* scheduling.

[4 marks]

(b) The listing on the next page gives code to implement a queue of objects as a circular buffer within an array of fixed size. Method put adds an element to the back of the queue, and get draws one from the front. The queued data occupies a region of the array, which moves further back as objects arrive and leave the queue. When it reaches the end of the array, it wraps around back to the beginning.

i. Explain why this class is not safe to use in a multi-threaded program. State the problem, and what kind of errors it can cause.

[4 marks]

ii. Give a specific example of how things could go wrong in Queue, describing the execution path of individual threads.

[6 marks]

iii. One way to make this class thread-safe is to mark some methods as **synchronized**. What is special about synchronized methods?
State exactly which methods in class Queue must be synchronized to make the code thread-safe.

[6 marks]

iv. A further limitation of the Queue class is that it does not check to see if the buffer is already full or empty before attempting a put or get operation. Improving this requires making Queue a *monitor*, so that when trying to add an object to a full queue, it waits until some space is available; and if a thread tries to fetch from an empty queue, then it blocks until a value arrives.
Using the Java wait and notifyAll methods, write out code for new put and get methods that implement this.
(You may ignore the possibility of threads being interrupted.)

[10 marks]

[Continued on next page.]

[Continued from previous page.]

```
public class Queue {

    private Object[ ] q;        // The array storing the queue

    private final int max;     // Maximum size of queue
    private final int top;     // Last slot in array

    private int low, high;     // Pointers to front and back
    private int size ;         // Current length of queue

    public Queue(int maxSize) {
        q = new Object[maxSize];
        max = maxSize;         // Record largest possible queue
        top = maxSize-1;       // and index of last slot in array

        high = 0; low = 0;     // Set initial values
        size = 0;              // for empty queue
    }

    public boolean isFull()  { return size == max; }

    public boolean isEmpty() { return size == 0; }

    public void put(Object o) {
        q[high] = o;           // Place new value at back of queue
        size = size+1;         // Queue is now bigger
        high = bump(high);     // Move upper pointer
    }

    public Object get() {
        Object o = q[low];     // Fetch item at front of queue
        size = size-1;         // Queue is now smaller
        low = bump(low);       // Move lower pointer
        return o;
    }

    // Increase pointer into circular buffer by one place
    private int bump(int i) {
        if (i >= top)          // When at the end of the array,
            return i-top;      //   loop around
        else
            return i+1;
    }
}
```